Exercise Sheet 4

1. Show that, to leading order, Cholesky factorization of an $n \times n$ matrix requires $n^3/3$ flops.

2. Compute a reduced QR factorization of the matrix

$$A = \begin{bmatrix} 3 & 7 \\ 0 & 12 \\ 4 & 1 \end{bmatrix}$$

   using Gram-Schmidt orthogonalization. Then extend it to a full one.

3. Use the Householder method to compute a full QR factorization of the matrix A of Exercise 2.

4. Experimentally verify backward stability of the Householder method in the following way.

   (a) Generate a random 50×50 matrix A with *known* QR factorization. In Matlab and Octave this can be achieved with the following commands.

```
R = triu(randn(50));
[Q,~] = qr(randn(50));
A = Q*R;
```

   Note that the default implementation of the QR factorization (command `qr` in the second line) uses the Householder method. In Python the same can be achieved with the following lines.

```
import numpy as np
R = np.triu(np.random.rand(50,50))
Q = np.linalg.qr(np.random.rand(50,50))[0]
A = Q@R
```

   (b) Next compute the QR factorization of A with the command

```
[Q2,R2] = qr(A);
```

   in Matlab and Octave or with

```
Q2,R2 = np.linalg.qr(A)
```

   in Python.

   (c) Now compare the relative forward and backward errors of this QR factorization. Interpret your findings. What do they tell you about the condition of the problem?

Upload your program via Moodle.

5. Write a program that
   (a) takes as input a full rank matrix $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$, where $m \geq n$,
   (b) computes the QR factorization of A using the Householder method,
   (c) returns the solution of $Ax = b$, if $m = n$, and the least squares solution otherwise.

   Upload your program via Moodle.

6. In Matlab (and Octave) linear systems of equations can be solved with the command `A\b` or, equivalently, `mldivide(A,b)`. When entering one of these commands Matlab decides which algorithm to apply. In the section "Algorithm for Full Inputs" of

   `https://www.mathworks.com/help/matlab/ref/mldivide.html`

   there is a flow chart showing how this decision making is done for *full* matrices. Discuss those parts of this flow chart that have been treated in the lecture.